Notenprogramm für Schülerinnen und Schüler

Kontext:

Die Noten stehen fest und die Zeugnisse werden demnächst in Druck gegeben. In diesem Schuljahr haben Sie leider zu spät bemerkt, dass ihr Notenschnitt schlechter ausfällt als gedacht. Um im nächsten Halbjahr einen guten Überblick zu behalten schreiben Sie ein Programm, welches die Berechnung der Noten und weitere Aufgaben automatisch erledigt. Zu den Aufgaben des Programms sollen die folgenden Features gehören:

- Man kann maximal 12 Fächer anlegen
- Man kann neue Fächer anlegen
- Jedes Fach besitzt einen Namen und ein Notenziel
- Man kann für ein Fach die Differenz zwischen Notenziel und aktuellem Notendurchschnitt berechnen lassen
- Man kann für ein Fach ein Notenziel eingeben
- Man kann für ein Fach mündliche und schriftliche Noten eingeben
- Man kann sich in einem Fach den Notendurchschnitt ausgeben lassen
- Man kann sich in einem Fach eine aktuelle Tendenz ausgeben lassen
- Es können pro Fach maximal 20 Noten gespeichert werden
- Die Noten werden im Punkteschema gespeichert
- Die schriftlichen Noten zählen 50% der gesamten Note

Hinweis: Auch wenn Sie sehr schnell auf mögliche Erweiterungen stoßen, planen Sie diese nicht ein, sondern konzentrieren Sie sich auf die gestellten Anforderungen.

Aufgaben:

1. Wenden Sie die Abbott-Methode an um mögliche Objektkandidaten herauszufinden.
2. Erstellen Sie mit Hilfe des Tools Objektdiagramm einen ersten Softwareentwurf in Form eines Objektdiagramms indem Sie die identifizierten Objekte mit möglichen Werten ausstatten und auch die Verbindungen zwischen den Objekten angeben. Verbessern Sie ihren Entwurf aufgrund weitere möglicher Szenarien.
3. Entwickeln Sie aus dem Objektdiagramm heraus ein passendes Entwurfsdiagramm. Verbessern Sie ihren Entwurf mit Hilfe weiterer Szenarien.
4. Überführen Sie das Entwurfsdiagramm in ein passendes Implementationsdiagramm.
5. Implementieren Sie die Klassen und Methoden, die das Implementationsdiagramm vorgibt und testen Sie ihre Methoden mit Hilfe von Unit-Tests.

Museumsverwaltung

Kontext:

Ein kleines Museum hat eine Software in Auftrag gegeben, womit die Exponate besser verwaltet werden können. Die Exponate können dabei unterteilt werden in Bilder, Skulpturen und Artefakte, die jeweils entweder gerade ausgestellt sind oder nicht. Künstler und Forscher können dem Museum neue Exponate zur Verfügung stellen. Gespeichert werden dann Alter, Künstler und Herkunft. Andere Museen können auch Exponate ausleihen, wobei dann das ausleihende Museum und das Ende der Leihe gespeichert werden sollen. Insgesamt hat das Museum mit seiner Ausstellungsfläche (100 Exponate) und seinem Lager (300 Exponate) eine Gesamtkapazität von 400 Exponaten. Folgende Funktionen soll das Programm zur Verfügung stellen:

- Es müssen neue Exponate gespeichert werden können.
- Es müssen vorhandene Exponate ausgeliehen werden können.
- Es soll ausgegeben werden können, wie viele Exponate gerade in der Ausleihe, im Lager oder in der Ausstellung sind.
- Es soll ausgegeben werden können, wie viele Exponate von einem bestimmten Typ in der Ausstellung sind.

Hinweis: Auch wenn Sie sehr schnell auf mögliche Erweiterungen stoßen, planen Sie diese nicht ein, sondern konzentrieren Sie sich auf die gestellten Anforderungen.

Aufgaben:

1. Wenden Sie die Abbott-Methode an um mögliche Objektkandidaten herauszufinden.
2. Erstellen Sie mit Hilfe des Tools Objektdiagramm einen ersten Softwareentwurf in Form eines Objektdiagramms indem Sie die identifizierten Objekte mit möglichen Werten ausstatten und auch die Verbindungen zwischen den Objekten angeben. Verbessern Sie ihren Entwurf aufgrund weitere möglicher Szenarien.
3. Entwickeln Sie aus dem Objektdiagramm heraus ein passendes Entwurfsdiagramm. Verbessern Sie ihren Entwurf mit Hilfe weiterer Szenarien.
4. Überführen Sie das Entwurfsdiagramm in ein passendes Implementationsdiagramm.
5. Implementieren Sie die Klassen und Methoden, die das Implementationsdiagramm vorgibt und testen Sie ihre Methoden mit Hilfe von Unit-Tests.

Sportverein - Mitgliederverwaltung

Kontext:

Ein kleiner Sportverein hat eine Software in Auftrag gegeben, womit die Mitglieder besser verwaltet werden können. Die Mitglieder können dabei unterteilt werden in Aktive und Passive. Zu den Aktiven zählen Spieler, Trainer und die Koordinatoren. Zu den Passiven gehören Sponsoren und Fans. Alle Mitglieder erhalten bei Anmeldung eine VereinsID, welche eindeutig vergeben wird. Außerdem wird, zur Berechnung der Ehrungen, noch das Anmeldedatum gespeichert und natürlich der Name, Vorname und die Adresse. Bei den Sponsoren wird noch das monatliche Sponsoring gecpeichert. Da die aktiven Mitglieder für ihre Tätigkeiten im Verein eine Aufwandsentschädigung bekommen, wird diese Eigenschaft nur bei den aktiven gespeichert. Außerdem können bei den Mitgliedern die Spieler sind noch die Eigenschaften Position, Tore und Einsätze gespeichert werden.

Folgende Funktionen soll das Programm bereit stellen:

- Es müssen neue Mitglieder aufgenommen werden können.
- Es müssen Mitglieder gelöscht werden können.
- Aufgrund der Anzahl der Einsätze sollen die Spieler eine Auflaufprämie in Höhe von 50€ pro Spiel gutgeschrieben bekommen.
- Die Summe der monatlichen Gelder aller Sponsoren soll aufsummiert zurückgegeben werden können.
- Es soll abgeglichen werden können ob die Summe der Sponsorgelder die der Ausgaben übersteigt.

Hinweis: Auch wenn Sie sehr schnell auf mögliche Erweiterungen stoßen, planen Sie diese nicht ein, sondern konzentrieren Sie sich auf die gestellten Anforderungen.

Aufgaben:

1. Wenden Sie die Abbott-Methode an um mögliche Objektkandidaten herauszufinden.
2. Erstellen Sie mit Hilfe des Tools Objektdiagramm einen ersten Softwareentwurf in Form eines Objektdiagramms indem Sie die identifizierten Objekte mit möglichen Werten ausstatten und auch die Verbindungen zwischen den Objekten angeben. Verbessern Sie ihren Entwurf aufgrund weitere möglicher Szenarien.
3. Entwickeln Sie aus dem Objektdiagramm heraus ein passendes Entwurfsdiagramm. Verbessern Sie ihren Entwurf mit Hilfe weiterer Szenarien.
4. Überführen Sie das Entwurfsdiagramm in ein passendes Implementationsdiagramm.
5. Implementieren Sie die Klassen und Methoden, die das Implementationsdiagramm vorgibt und testen Sie ihre Methoden mit Hilfe von Unit-Tests.